Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments

Conference Paper · October 2012

DOI: 10.1109/CRISIS.2012.6378942

Siwar Kriaa

Ecole Centrale Paris

Marc Bouissou

Électricité de France (EDF)

Modeling the Stuxnet Attack with BDMP: Towards More Formal Risk Assessments

Siwar Kriaa
Grenoble Institute of Technology
Grenoble 38402, France
Email: siwar.kriaa@gmail.com

Marc Bouissou
École Centrale de Paris
Châtenay-Malabry 92295, France
Email: marc.bouissou@ecp.fr

Ludovic Piètre-Cambacédès
Electricité de France (EDF) R&D
Clamart 92141, France
Email: ludovic.pietre-cambacedes@edf.fr

***Abstract*—Attack modeling has recently been adopted by security analysts as a useful tool in risk assessment of cyber-physical systems. We propose in this paper to model the Stuxnet attack with BDMP (Boolean logic Driven Markov Processes) formalism and to show the advantages of such modeling. After a description of the architecture targeted by Stuxnet, we explain the steps of the attack and model them formally with a BDMP. Based on estimated values of the success probabilities and rates of the elementary attack steps, we give a quantification of the main possible sequences leading to the physical destruction of the targeted industrial facility. This example completes a series of papers on BDMP applied to security by modeling a real case study. It highlights the advantages of BDMP compared to attack trees often used in security assessment.**

Keywords: Stuxnet, cyber-physical systems, BDMP, security modeling, risk assessment, attack trees.

I. Introduction

Since 2010, the Stuxnet worm has been of particular interest for the media and security experts. Not only because of its high degree of sophistication but also because it targeted the control systems of an industrial installation and led, for the first time, to major physical damage. It has since been a trigger to urge industries to protect their critical infrastructures against cyber-attacks and to pay more attention to interdependencies between the cyber and the physical parts of their systems. Many studies explored Stuxnet with more or less details and gave technical explanations about the infiltration and the propagation of this worm into the core network and the control system [1], [2]. Yet, very few provided a model enabling a global understanding of the attack. Modeling an attack is a paramount step in the procedure of securing a cyber-physical system for several reasons [3]. First, it enables the identification of the weaknesses and the different access points of the system and makes the attack vectors more evident. Secondly, it makes the search for efficient solutions to mitigate these vulnerabilities easier. Finally, it helps understanding the behavior of the attacker and assessing the effect on the physical infrastructure. The only existing models of the Stuxnet attack are based on attack trees [4] or graphs [5]. We propose in this paper to model Stuxnet with the BDMP formalism [6] in order to: i) better reflect the dynamics of this assault, ii) enable a coarse quantification of the attack success probability and finally, iii) highlight the advantages of BDMP compared to existing models.

Our paper will be organized as follows: Section II gives a global overview of the Stuxnet attack. Section III introduces the BDMP formalism and its modeling objects. Section IV presents the BDMP model of Stuxnet and gives risk assessment results. Section V yields a comparison between the BDMP model and existing ones. Finally, Section VI concludes the paper and proposes further work and improvements.

II. Stuxnet Attack Overview

Stuxnet ultimately targeted supervisory, control and data acquisition systems (SCADA) running a Windows environment that hosts specific Siemens industrial control systems (namely the WinCC, PCS7 and STEP7 platforms) and connected to specific types of Programmable Logic Controller devices (PLCs). It reprograms PLCs in a way that modifies the system operation leading to damage to the physical infrastructure under control. The Stuxnet attack affected mainly Iranian nuclear enrichment facilities and resulted in slowing down the production of centrifugal machines and finally damaging them. The sophistication of the malware and the very specific systems it targeted led to the conclusion that such attack could not be developed by a group of persons but rather by a nation-state.

Considering the sensitivity of the facility targeted by Stuxnet, its SCADA system was not directly connected to the Internet (and presumably non-industrial networks of the facility). Consequently, the best attack path for Stuxnet was to compromise an external device, typically a USB thumb drive, that would be later connected to the control system. So, the first step of the attack was to propagate throughout the Information System of the Enterprise corporate network to increase the probability of reaching the industrial network. To reach this goal, Stuxnet exploits several Windows' vulnerabilities and at least four 0-day exploits [2]. Another specificity of this malware is that it injects its entire payload into other legitimate processes and use several rootkits to escape detection. We give more technical details about Stuxnet dynamics and its life cycle in Section IV-B.

III. BDMP and Security

A. Short presentation of BDMP

BDMP are a graphical modeling formalism initially conceived for safety and reliability assessment [7]. This formalism is a combination of classical fault trees with Markov processes. It consequently provides a good readability and hierarchical representation like fault trees as well as advanced quantification capabilities. Unlike static fault trees, the BDMP formalism enables modeling dynamic features with a special type of link called "triggers". BDMP model the different combinations of events (leaves of the tree) that may lead to the undesired event (root of the tree) which can be for example the system failure. Each leaf is associated to a "triggered Markov process" (see §III-B1) that models the different states of the leaf. Moreover, BDMP have very interesting mathematical properties. They allow a dramatic reduction of combinatorial problems in operational applications, especially when they are processed using a method based on sequence exploration. This method also gives very interesting qualitative results including the most probable sequences that lead to the undesirable event.

B. The BDMP formalism applied to security

The BDMP formalism has recently been adapted to the security field [8], [6], [9]. New security leaves have been defined to model attack steps or in some cases security events. An attack leaf can be either in "Idle" or "Active" mode. The former is used when nothing is in progress; the latter models an on-going event, generally an attack event in progress. Detection and reaction aspects can also be modeled. A complete software workbench is available to build and analyze the model in [9].

1) The elements of a BDMP: A security-oriented BDMP $\{\mathcal{A}, r, T, P\}$ is made of: a multi-top coherent attack tree A, a main top event r of A, a set T of triggers and a set P of "triggered Markov processes" P_i associated to the leaves of A. The Markov process P_i is said to be "triggered" because it switches instantaneously from one of its modes to the other one according to the state of some externally defined Boolean variable, called "process selector". An important feature of BDMP is the concept of "relevant event". An event is said to be non relevant if its realization does not have any effect on the realization time of the top event. For example if one leaf of an "OR" gate is realized, other leaves of the gate are no longer relevant because the gate is realized. Non relevant processes are trimmed during the processing when exploring the possible sequences. Trimming strongly reduces the combinatorial explosion while yielding exact results in our assumptions. The process selectors are defined by means of triggers. A trigger, graphically represented with a red dashed arrow, can modify the mode of the processes associated to the leaves of the sub-tree it points to, when the event that is the origin of the trigger changes from FALSE to TRUE (or conversely). The complete definition of the semantics of a BDMP can be found in [7], [6].

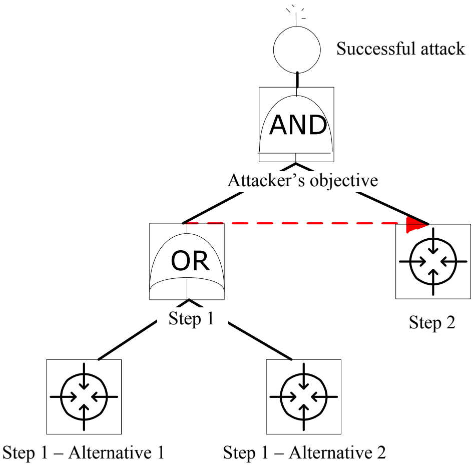

Figure 1. Example of a simple BDMP

Fig. 1 represents a very simple BDMP modeling a two step attack with two alternatives for Step 1. The "trigger" ensures that the leaf representing Step 2 is realizable only if Step 1 has been completed. The times needed for the realization of the leaves are defined by stochastic processes; their behaviors can be made dependent on other leaves by means of the triggers.

2) Modeling objects:

- BDMP leaves for attack modeling

Attack leaves define the different types of events that we can consider in an attack scenario. The three kinds of leaves are described in Tab. I. Their complete description can be found in [6].

In addition to the λ and γ parameters associated to the attack, detection and reaction parameters have been added, to cover defensive aspects [6].

- Gates and links

The BDMP models use classical logic gates "AND" and "OR". More specific gates (e.g. "PAND" "Aggregate OR") are defined in the documentation associated to the modeling tools [9]. In addition to classical logic links, BDMP models introduce other specific links. Tab. II describes two of them, the "Trigger link" and the "Before link", that are used in our model.

IV. Stuxnet modeling with BDMP

A. Network architecture of the industrial site

Several security consulting services published detailed information about the components of the targeted Siemens platforms and typical network architectures [5]. Based on these studies, we have defined a simplified architecture of what could have been the targeted one. It is represented

Table I
Basic BDMP leaves for security modeling

Representation	Modeled behavior
⌖	The "Attacker Action" (AA) leaf models an attacker step towards the accomplishment of his objective. The Idle mode means that the attacker has not yet at this stage tried to do this action. The Active mode corresponds to actual attempts for which the time needed to succeed is exponentially distributed with a parameter λ. The Mean Time To Success (MTTS) for this action is equal to $1/\lambda$.
TSE ⌖	The "Timed Security Event" (TSE) leaf models an event the realization of which is necessary for the attack success but that is not under the direct control of the attacker. The time needed for its realization is exponentially distributed (MTTS=$1/\lambda$). If the leaf comes back to the Idle mode, the leaf state can then be either Realized or Not Realized, depending on whether the TSE occurred or not in Active mode.
ISE! ⌖	The "Instantaneous Security Event" (ISE) leaf models a security event that can happen instantaneously with a probability γ when the leaf switches from the Idle mode to the Active mode. In the Idle mode, the event cannot occur and the leaf stays in the state Potential. In the Active mode, the event is either Realized or Not Realized.

Table II
Special BDMP links

Representation	Modeled behavior
— — — — → Red dashed arrow	"Trigger Links" define the dynamic aspect of BDMP. The element pointed by the trigger link is not activated until the realization of the origin gate/leaf of the trigger. When this element becomes activated, it transmits the activation signal it receives from its parents to the sub tree targeted by the trigger.
- - - - - - - - → Blue dotted arrow	"Before Links" link only ISE leaves. They define the order in which the corresponding instantaneous security events are realized (or not).

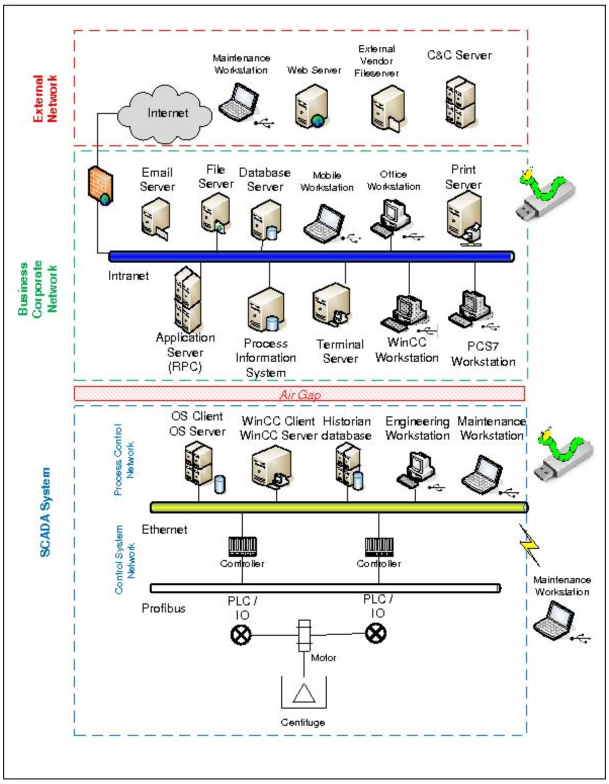

Figure 2. Facility network architecture of an industrial site

in Fig. 2 and will be the basis of our BDMP model of the Stuxnet attack in Section IV-C.

The whole facility architecture is composed of two main security zones, the Business Corporate Network and the SCADA system.

The Business Corporate Network hosts the Enterprise usual Information System. It comprises servers and workstations that enable classical daily applications (emails, reporting, accountability...), the Enterprise Resource Planning (ERP) system, etc. It may also host WinCC SQL Server databases that provide high level information to end users and store STEP7 project files, as well as applications that manage PCS7 or WinCC projects. Data can be exchanged between terminals via a local area network that hosts local databases and process information servers. This network can exchange data with external networks connected to the Internet through a “demilitarized zone". The communication is protected by firewalls and other security modules.

The SCADA system includes a Process Control Network and a Control System Network. The Process Control Network consists of WinCC and PCS7 clients and servers which are connected to PLCs and enable communication with them. WinCC machines provide HMI client/server systems used to monitor the industrial process and visualize messages and real-time data. PCS7 machines include basic data collection functions for project data, process values, archives, alarms and messages. PCS7 servers provide all process data and connect PLCs to the Process

Control Network [5]. The control system network includes WinCC/PCS7 servers and PLCs. It controls and supervises the physical process. PCS7/WinCC server and client can be installed on the same hardware which is the case in our model for simplification. PLCs send control signals via a Process Field Bus (Profibus) to speed regulators that control the rotation of motors. This network includes as well WinCC SQL Server databases and other engineering or maintenance workstations. We suppose that, for security reasons, the SCADA system is isolated by an air gap so that no network connection is possible between the two security zones.

B. Stuxnet dynamics

In this sub-section we give more details about the malware main phases and attack steps. Text written in `Typewriter` font denotes the corresponding leaves in the BDMP model given in Figure 3 and Appendix (not all leaves are mentioned). Basically, we can distinguish two main phases: 1) infiltration and propagation into the corporate Enterprise network; 2) compromising SCADA systems and industrial sabotage.

1) Infiltration and propagation into the corporate Enterprise network: We assume in our model that the whole network is initially non-infected. It is then very likely that the very first infiltration of the malware was introduced by infected removable media, which represent the main attack vector, into a workstation of the business corporate network. The user action of inserting an infected removable drive is modeled by a TSE leaf `user USB key execution` as it is dependent on the user and not under the control of the attacker. Stuxnet exploits two Windows vulnerabilities to spread to and from removable drives. One is Windows Shell LNK vulnerability linked to the system handling of shortcuts using ".LNK" and ".PIF" files. The other is autorun.inf file vulnerability which enables self-execution of the removable drive when inserted. The user has just to open a compromised file folder on his USB drive to let the worm do the rest of the work. When the first step is realized, the malware instantaneously exploits one of the two vulnerabilities modeled by ISE leaves `Win LNK vuln` and `autorun.inf vuln` in order to infiltrate the system.

Once introduced into the system, the next attack step is self installation. The malware loads instantaneously the main dropper which is a dynamic link library (.dll) that contains Stuxnet functions, files and rootkits into a trusted process generally default Windows processes or executable files of security products installed. Then it, `checks Windows config`: it targets particularly 32 bits machines running the operating systems Windows XP/2003/Vista/7 or Windows Server. Next, it `checks admin rights` of the current user. If not found, the malware exploits one of two zero-days, `keyboard layout` and `task planner` vulnerabilities, to elevate its privileges.

The worm proceeds then to the execution of its main installer. It comprises two main steps: `install Win rootkit` in order to escape detection then updating the last version of the malware. To install the rootkit, Stuxnet loads a driver file legitimately signed by Realtek certificate and used to scan the main filesystem driver objects. It then creates a new device object and attaches it to the driver chain of the previous driver objects to be the first to receive requests to/from these drivers. This allows the malware to filter out files with ".LNK" and ".TMP" extensions to hide malicious files.

To update the last version of the malware, Stuxnet can either establish a `P2P communication` by installing an RPC server on the infected machine and wait for connections from RPC clients or it can directly download the latest updates from Control and Command Server (`C&C server communication`). Stuxnet communicates with remote servers on port 80 via HTTP. It injects itself into Internet explorer or creates another browser process to bypass firewall security rules.

The malware tries to infect as many workstations as possible in the corporate business network to maximize its chances to transit later to the Control Network. The three main ways of infection, self-replication and propagation are: 1) `removable media` connected to compromised machines. 2) the `LAN`; through `network shares`, Print Spooler Service exploit, Windows Server Service exploit or connections to WinCC remote databases. In this last case, Stuxnet searches for WinCC environments. If found, it connects to the database using default Siemens password and sends malicious code via SQL queries. 3) WinCC/Step7 project files associated to WinCC SIMATIC Manager. Stuxnet searches for and infects files with extensions ".S7P", ".MCP" or ".TMP". It waits then for the user to open the infected file (`user opens file project`) to load its .dll file, decrypt data and execute infection routines.

2) Compromising SCADA systems and industrial sabotage: The next main step for Stuxnet is to reach and compromise the SCADA Network to attack the industrial system. The network being isolated from the Corporate Business Network for security reasons, the malware waits until it is somehow carried to the Process Control Network by an employee connecting infected removable drives or by a maintenance workstation (`infection of a control PC`). Stuxnet looks first for WinCC/Step7 software on the control PC used to configure the PLC. If found, it installs a rootkit: it loads a library file (s7otbx**d**x.dll) used for the communication between the control PC and the PLC, renames it (s7otbx**s**x.dll) and inserts malicious code into the new file. After checking connection to PLC as well as other specific configuration (PLC model, Profibus configuration, speed regulators number), the malware proceeds to infecting and modifying PLC function blocks. The code executed on PLC differs depending on its CPU type; only 6ES7-315-2 or 6ES7-417 modules are targeted. `Flag sys 300` and `Flag sys 400` enable to choose one of these two options. In the case of 300-series systems, the malware collects data for a period going from 13 days to 3 months before sending falsified data to motors on the communication bus for around 50 minutes. For 400-series systems, the code sequence is more complex. Coarsely, it intercepts input and output signals of PLC and provides false data to the logic code sequence in order to falsify output returned signals (man-in-the-middle attack). The malware operates without being detected or inducing any suspicious signals or abnormal values to be visualized in return to operators.

Through the falsified PLC output signals, the attacker gives instructions to motors to alternate high then low frequency rotation. This phase can last several months causing the physical materiel to wear down slowly and consequently worsen its performance. It can even end into machines self-destruction.

C. Stuxnet model

We model in Fig. 3 the top part of the Stuxnet attack BDMP with its main phases: infiltration, self-installation and attack of the industrial system. The BDMP of the last two phases are detailed in Appendix (Fig. 6 and Fig. 7). The sub-tree modeling the different propagation paths of the malware (Fig. 6) is not required later in risk quantification because the attack can succeed from the first chance; that is why it is not linked to the top event of the BDMP.

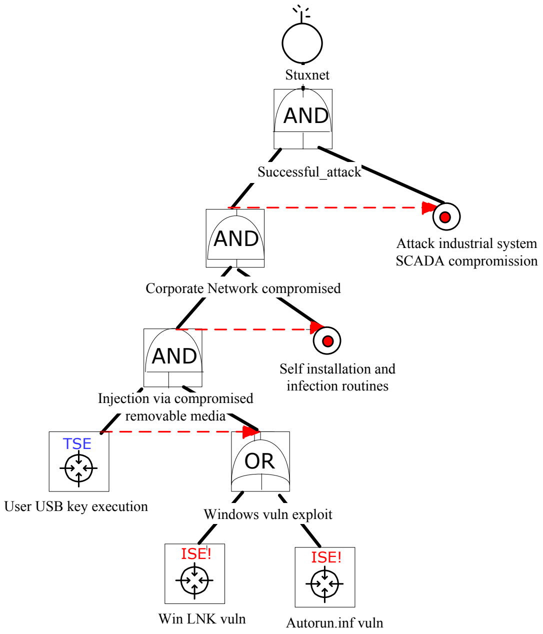

Figure 3. Top part of the Stuxnet model

In this BDMP model, we can parameterize the different leaves following their formal specification by estimating success rates and probabilities (λs and γs). Such parameterization is used later in the quantitative analysis: it enables the computation of the attack success probability as well all possible sequences that lead to the attack success. We list in Tab. III parameter values corresponding to attack leaves that we have chosen in the BDMP model based on our own estimation and writings by security consultants [10].

Table III
Parameters of the use case

Subtree	Leaf label	Parameter
Infiltration	user USB key exec	λ=5.787e-6 (MTTS= 2 days)
	Win LNK vuln autorun.inf vuln	γ=1/2
Installation	Admin rights	γ=0.7
	keyboard layout vlun task planner vuln P2P communication CC server communication	γ=1/2
SCADA compromising	infection of a control PC	λ=7.7e-7 (MTTS= 15 days)
	collect data	λ=3.86e-7 (MTTS= 1 month)
	user opens file project	λ=1.16e-5 (MTTS= 1 day)
	PLC sends false data to motors	λ=3.33e-4 (MTTS= 50 min)
	intercept in out PLC signals	λ=8.36e-7 (MTTS= 1 month)
	modify out signals	λ=1.15e-5 (MTTS= 1 day)
Propagation	removable media	λ=5.79e-6 (MTTS= 2 days)
	network shares	λ=1.39e-4 (MTTS= 2 hours)
	print servers vuln	λ=9.25e-5 (MTTS= 3 hours)
	service server RPC vuln	λ=2.77e-4 (MTTS= 1 hour)
	cascade centrifuges	γ=0.1 per centrifuge
For all other ISE leaves: γ=0.99 (almost sure)		

D. Quantitative and qualitative risk analysis

BDMP enable not only attack representation but yield also quantitative and qualitative results directly useable for risk assessment. KB3 quantification tools [9] enable BDMP analysis namely the enumeration of all possible attack paths ordered by their probabilities of occurrence and contributions to the final attack success. We summarize in Tab. IV all possible attack sequences; vertically readable. We number steps that are not in common for all sequences and we denote in Tab. V each sequence by the numbers of the steps that differentiate it from other sequences. According to [10], Stuxnet attacked a group of cascades of centrifuges, each cascade comprises 164 centrifuges. We suppose, in our BDMP model, that the attack is successful when at least 3 centrifugal machines fail. This hypothesis is not important (at least from a qualitative point of view): when Stuxnet infects the PLC, it can make all the machines fail. It is just a matter of time. The succession of failures of a set of identical components is represented by the leaf at the extreme right of the BDMP "`cascade centrifuges`". The behavior of such a leaf is depicted in Fig. 4. Each state is defined by the number of failed components.

Table IV
List of successful attack sequences (NR: Not Realized)

<table>
<tr><td colspan="4">User USB key execution</td></tr>
<tr><td>1-Win LNK vuln
autorun.inf vuln (NR)</td><td>2-Win LNK vuln(NR)
autorun.inf vuln</td><td colspan="2">3-Win LNK vuln
autorun.inf vuln</td></tr>
<tr><td colspan="4">self injection into process</td></tr>
<tr><td colspan="4">check Windows config</td></tr>
<tr><td rowspan="2">4-Admin rights</td><td colspan="3">Admin rights (NR)</td></tr>
<tr><td>5-keyboard layout vuln
task planner vuln(NR)</td><td>6-keyboard layout vuln(NR)
task planner vuln</td><td>7-keyboard layout vuln
task planner vuln</td></tr>
<tr><td colspan="4">load driver legitimately signed</td></tr>
<tr><td colspan="4">scanning filesystem drivers</td></tr>
<tr><td colspan="4">new device object attachment</td></tr>
<tr><td colspan="4">filter out .lnk .tmp files</td></tr>
<tr><td colspan="2">8-C&C server communication</td><td colspan="2">9-P2P communication</td></tr>
<tr><td colspan="4">infection of a control PC</td></tr>
<tr><td colspan="4">check STEP7 or WinCC</td></tr>
<tr><td colspan="4">Load library</td></tr>
<tr><td colspan="4">rename replace library</td></tr>
<tr><td colspan="4">check PLC exists</td></tr>
<tr><td colspan="4">Check PLC model</td></tr>
<tr><td colspan="4">check Profibus config</td></tr>
<tr><td colspan="4">check speed regulators number</td></tr>
<tr><td colspan="4">modify PLC function blocks</td></tr>
<tr><td colspan="4">rootkit400 activated</td></tr>
<tr><td colspan="4">intercept in out PLC signals</td></tr>
<tr><td colspan="4">modify out signals</td></tr>
<tr><td colspan="4">fail cascade centrifuges</td></tr>
</table>

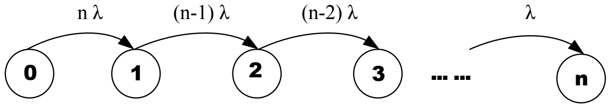

Figure 4. Markov chain modeling the failure process of a set of n identical components (n=164 in the case study)

Using KB3 quantification tools [9] we compute the probability of each possible sequence and its contribution to the overall probability of the attack success which is estimated to 0.6 for a 400-series system, with the chosen parameters. The results are given in Tab. V.

We can for example infer from these quantitative results that it is more probable for the malware to update itself through communicating with the Control and Command server that enables the attacker acting remotely on the infected system rather than waiting for RPC clients to connect for Peer-to-Peer communication. We can also notice, but it seems more obvious, that the attack is more likely to succeed when administrator rights are available on the system.

Fig. 5 plots the evolution of attack success probability according to time. We can see that success probability increases by time and reaches an asymptote after around 6 months. This asymptote is the global attack success probability; equal to 0.6 in the case of a 400-serie PLC. This probability never reaches 1 because we consider all attack failure cases including targeted configuration not found and PLCs not connected to infected configuration machines. The evolution of success probability is also tightly linked to the realization of long-phased attack steps

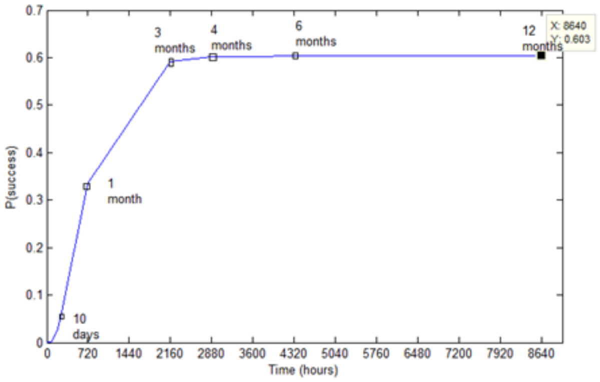

Figure 5. Attack success probability according to time

mainly at `Attack industrial system` phase.

“Stuxnet was discovered in July 2010, but is confirmed to have existed at least one year prior and likely even before"[1]. The value of 6 months that we obtain by our quantification has the same order of magnitude but is clearly inferior for the following reasons: (i) we chose rather high values for the model’s parameters (ii) the malware was discovered after the intended effects had taken place.

V. Advantages and limits of our approach

A. Comparison with other Stuxnet existing models

An attack tree [11] modeling Stuxnet with our hypotheses would be very close to the BDMP without its triggers and precedence (before) links. The attack tree model found in [4] is a good illustration. It gives no indication about the constraints on the order of attack steps. Therefore, it

Table V
Quantification results

Sequences	1-4-8 2-4-8 3-4-8	1-4-9 2-4-9 3-4-9	1-5-8 2-5-8 3-5-8	1-6-8 2-6-8 3-6-8	1-7-8 2-7-8 3-7-8	1-5-9 2-5-9 3-5-9	1-6-9 2-6-9 3-6-9	1-7-9 2-7-9 3-7-9
Proba per seq	1.06e-1	4.54e-2	1.14e-2			4.86e-3		
Contrib per seq	17.65%	7.56%	1.89%			0.8%		
Sum of contrib	52.95%	22.68%	17.01%			7.2%		

is useless without a detailed textual explanation. On the contrary, once a reader has in mind the correspondence between the necessarily short names of the leaves and the real events, he has a complete description of the attack with the BDMP. This modeling power is what makes the risk quantification of Section IV-D possible. With an attack tree, the only possible quantification corresponds to the assumption that all timed transitions are enabled from the start of the attack. This gives grossly erroneous results. This drawback of attack trees is particularly obvious in the case of Stuxnet where many attack steps are highly dependent from one another.
The attack graph found in [5] is only a visual representation with no quantification capabilities. Moreover, its non hierarchical and cyclic structure makes it less readable than a tree structure. It focuses on the propagation aspects of the worm, covering only a subpart of our model.

B. Limits of the model

There are of course several limits related to our work. Firstly, the quantification results cannot be considered very accurate as the parameters of attack leaves depend strongly on the real circumstances and the assumptions on the network architecture. More generally, as discussed more in details in [8], the modeling of security steps with exponential distributions and their parameterization are less reliable than in dependability studies. Other distributions might have been used, and their parameters more thoroughly evaluated. Moreover, in addition to the macroscopic sequence analysis and global success probability estimation found in IV-D, sensitivity analyses could have completed our analysis (their principle is explained in [9]) and helped comparing countermeasures. Such analyses could not be reflected here mainly for space reasons. Finally, from a more fundamental standpoint, BDMP are not well-suited to model cyclic behaviors of the malware.

VI. Conclusion and perspectives

In this paper we modeled the fundamental mechanisms of the Stuxnet attack in a unique and rigorous graphical representation, and gave quantification results for each possible attack sequence. These results reflect the potential of BDMP for modeling the steps of an attack and its global progress. They offer quantification tools to enumerate possible attack sequences with their probabilities and contributions to the overall attack success. This model also enables attack vectors and access points that an attacker may exploit in order to infiltrate a system and take control over the main control functions to be represented. Furthermore, we have modeled the attack *a posteriori*. It should be noted that the BDMP formalism is also an efficient tool to model attacks *a priori*, capturing the different potential scenarios for a given attacker objectives, and offering a panoramic view on all possible sequences leading to the attack success. In the framework of studying security of cyber-physical system, we intend to expand the functionalities of this formalism in order to model more sophisticated attack behaviors and assess their impact on the physical infrastructure. We will also work on adapting the BDMP model to capture and quantify the effects of security measures on the system safety and vice versa.

References

[1] N. Falliere, L. O. Murchu, and E. Chien, “W32.Stuxnet Dossier (v1.4),” Symantec report, Feb. 2011.

[2] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, “Stuxnet under the microscope (v1.0),” ESET, pp. 1–85, Feb. 2011.

[3] G. Dondossola, L. Piètre-Cambacédès, J. McDonald, M. Ekstedt, and a. Torkilseng, “Modelling of cyber attacks for assessing smart grid security,” in *2011 CIGRE D2 Colloquium*, Buenos Aires, Argentina, Oct. 2011.

[4] J. R. Nielsen, “Evaluating information assurance control effectiveness on an air force supervisory control and data acquisition (SCADA) system,” Master’s thesis, Air Force University, 2011.

[5] Tofini Security, Abterra Technologies, and ScadaHacker.com, “How stuxnet spreads, a study of infection paths in best practice systems (v1.0),” Whitepaper, Feb. 2011.

[6] L. Piètre-Cambacédès and M. Bouissou, “Attack and defense dynamic modeling with BDMP,” in *Proc. of the 5th Int. Conf. on Mathematical Methods, Models, and Architectures for Computer Networks Security (MMM-ACNS-2010), LNCS 6258*, St Petersburg, Russia, Sep. 2010, pp. 86–101.

[7] M. Bouissou and J.-L. Bon, “A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes,” *Reliability Engineering & System Safety*, vol. 82, no. 2, pp. 149–163, Nov. 2003.

[8] L. Piètre-Cambacédès and M. Bouissou, “Beyond attack trees: dynamic security modeling with Boolean logic Driven Markov Processes (BDMP),” in *Proc. 8th European Dependable Computing Conf. (EDCC-8)*, Spain, Apr. 2010, pp. 199–208.

[9] L. Piètre-Cambacédès, Y. Deflesselle, and M. Bouissou, “Security modeling with BDMP: from theory to implementation,” in *Proc. of th 6th IEEE Int. Conf. on Network and Information Systems Security (SAR-SSI 2011)*, La Rochelle, France, May 2011, pp. 1–8.

[10] D. Helan, “Stuxnet: Analysis, myths and realities,” Actu Secu, Feb. 2011.

[11] B. Schneier, “Attack trees,” *Dr. Dobb’s Journal*, vol. 12, no. 24, pp. 21–29, 1999.

Appendix

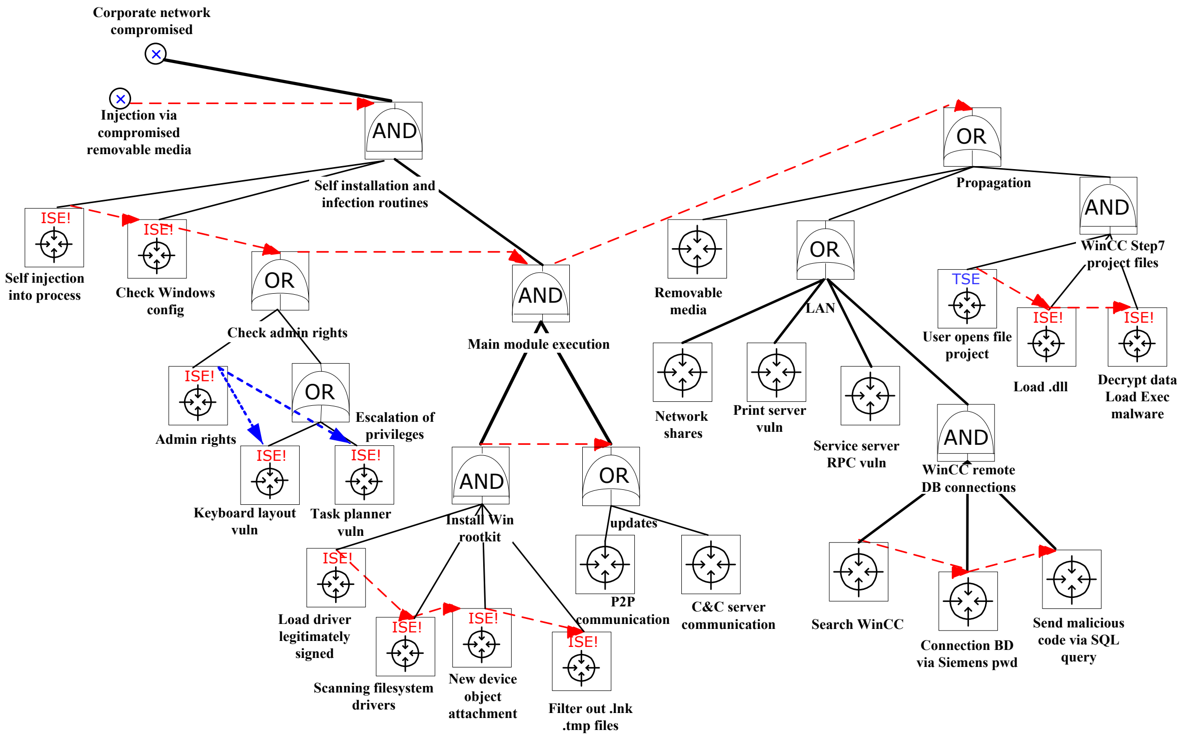

Figure 6. BDMP of the "self installation and infection routines" phase

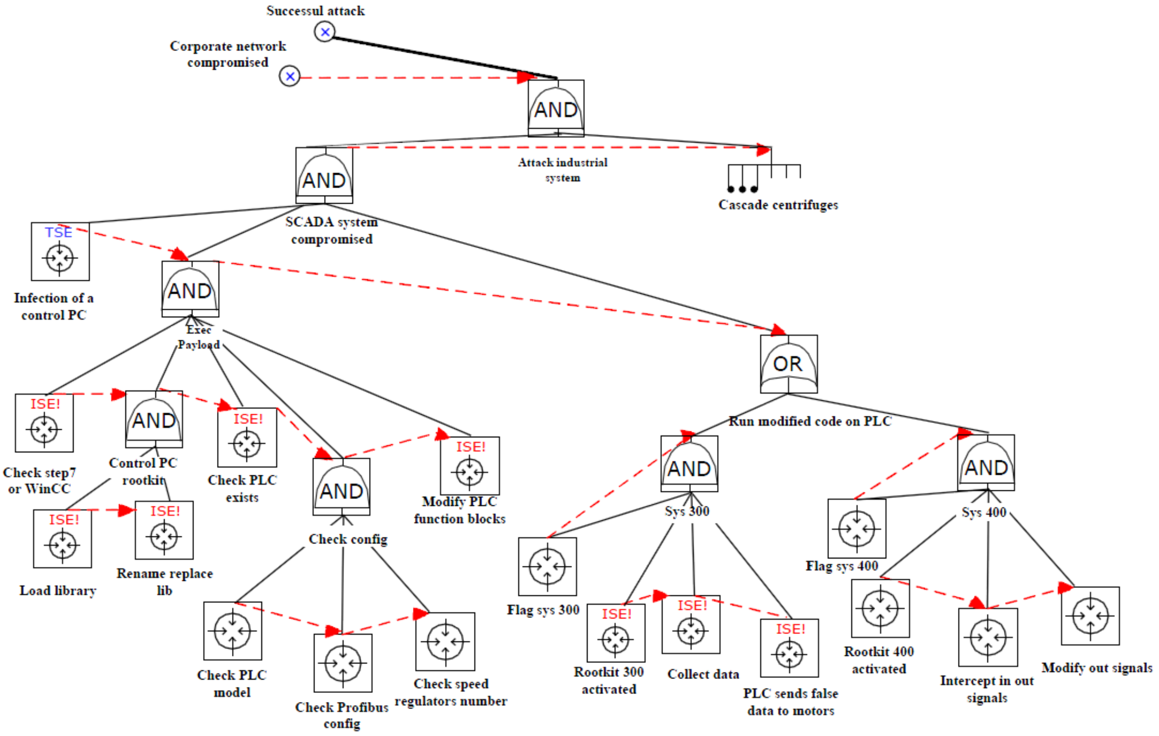

Figure 7. BDMP of the "attack industrial system" phase